Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-5 (canceled)

6. (original): A method of supporting conditional execution in a very long instruction word (VLIW) based array processor with subword execution, the method comprising:

providing general purpose flag bits (ACFs) that contain reduced condition information that is used for branching or conditional execution; and

specifying and setting a condition in ACFs based upon a condition code specification encoded in an instruction generating a condition.

7. (original): The method of claim 6 wherein instructions that execute conditionally do not affect the ACFs.

8. (original): The method of claim 6 wherein instructions that affect the ACFs execute unconditionally.

9. (original): The method of claim 6 further comprising the steps of:

executing a packed data instruction where the execution of each sub-word of the packed data operation is dependent upon the associated subword ACF.

Claims 10-48 (canceled)

49. (original): An indirect very long instruction word (VLIW) processing system comprising:

a first processing element (PE) having a VLIW instruction memory (VIM) for storing instructions in slots within a VIM memory locations;

a first register for storing a function instruction having a plurality of group bits defining instruction type and a plurality of unit field bits defining execution unit type;

a predecoder for decoding the plurality of group bits and the plurality of unit field bits; and

a load mechanism for loading the function instruction in an appropriate one of said slots in VIM based upon said decoding, the first processor further comprising:

at least two execution units, each execution unit receiving at least two operands from a register file;

each execution unit having instruction control lines derived from a registered instruction in a processor pipeline, the instruction control lines including conditional execution control lines to control conditional operation as specified in an instruction to be executed;

each execution unit producing a result and a latched arithmetic scalar condition state;

each execution unit having a first latch for holding the arithmetic scalar condition state for the instruction after the instruction has finished its execution state;

each execution unit having a second latch connected to the conditional execution control lines for holding instruction control signals for the instruction after the instruction has finished its execution state;

each execution unit having an arithmetic condition flag (ACF) generation unit for providing a Boolean combination of a present selected state with a previous state; and a single ACF latch for all of the execution units for storing the previous state and feeding the previous state back to the respective ACF generation unit.

50. (original): The system of claim 49 wherein the ACF latch is a programmer visible latch.

51. (original): The system of claim 49 wherein the PE further comprises a multiplexer connected to receive said Boolean combination from each of the ACF generation units and to controllably switch said Boolean combinations to branch logic in a sequence processor (SP).

52. (original): The system of claim 49 wherein the PE further comprises an arithmetic scalar flag (ASF) latch switchably connected to the output of each of the execution units.

Claims 53-55 (canceled)

56. (new): An apparatus for supporting conditional execution in a very long instruction word (VLIW) based array processor with subword execution, the apparatus comprising:

a processing element for providing general purpose flag bits (ACFs) that contain reduced condition information that is used for branching or conditional execution; and

an instruction having a condition code specification encoded therein, the processing element specifying and setting a condition in ACFs based upon the condition code specification generating a condition.

57. (new): The apparatus of claim 56 wherein instructions that execute conditionally do not affect the ACFs.

58. (new): The apparatus of claim 56 wherein instructions that affect the ACFs execute unconditionally.

59. (new): The apparatus of claim 56 further comprising the steps of:

executing a packed data instruction where the execution of each sub-word of the packed data operation is dependent upon the associated subword ACF.

60. (new): An indirect very long instruction word (VLIW) processing system comprising:

a fetch controller;

a VLIW instruction memory (VIM);

a processing element (PE) having a VIM controller, and a plurality of execution units, the VIM controller receiving a VLIW instruction from the fetch controller and generating VIM addresses for segmenting the VLIW instruction and storing the segmented VLIW instruction into slots within a VIM memory location, each slot corresponding to each execution unit;

each execution unit having a plurality of instruction control lines through which to load an instruction stored in the execution unit's corresponding slot, a portion of said plurality of instruction control lines carrying an instruction control signal for controlling conditional operation as specified in the loaded instruction, each execution unit receiving as input at least two operands and an arithmetic condition flag (ACF), the ACF representing a previous state of the processing element, each execution unit producing a result defining an execution state; and

an ACF latch connected to each execution unit, the ACF latch storing the previous state and feeding the previous state back to each execution unit.

61. (new): The indirect VLIW processing system of claim 60 wherein each execution unit producing an arithmetic scalar condition state.

62. (new): The indirect VLIW processing system of claim 61 wherein each execution unit further comprises a first latch storing the arithmetic scalar condition state.

63. (new): The indirect VLIW processing system of claim 62 wherein at least one of said plurality of execution units further comprises a second latch connected to the portion of said plurality of instruction control lines for holding the instruction control signal for the instruction after the instruction has finished its execution state.

64. (new): The indirect VLIW processing system of claim 63 wherein the at least one of said plurality of execution units further comprises an ACF generation unit for providing a Boolean combination of a present selected state with a previous state.

65. (new): The indirect VLIW processing system of claim 60 further comprising a plurality of processing elements, the fetch controller dispatching VLIW instructions to said plurality of processing elements.

66. (new): The indirect VLIW processing system of claim 60 wherein the ACF latch is a programmer visible latch.

67. (new): The indirect VLIW processing system of claim 61 wherein the PE further comprises an arithmetic scalar flag (ASF) latch switchably connected to the output of each of the execution units.

68. (new): The indirect VLIW processing system of claim 64 wherein the PE further comprises a multiplexer connected to receive said Boolean combination from each of the ACF generation units and to controllably switch said Boolean combinations to branch logic in a sequence processor (SP).

69. (new): A method of supporting conditional execution in a very long instruction word (VLIW) based array processor, the VLIW based array processor having a processor element, the method comprising:

receiving a VLIW instruction;

generating VIM addresses for segmenting the VLIW instruction and storing the segmented VLIW instruction into slots within a VIM memory location;

loading an instruction stored in one of the slots;

receiving an instruction control signal for controlling conditional operation as specified in the loaded instruction;

receiving as input at least two operands and an arithmetic condition flag (ACF), the ACF representing a previous state of the processing element;

producing a result based on the loaded instruction, said at least two operands and the ACF, the result defining an execution state of the instruction; and

storing the previous state of the processing element for a subsequent producing step.

70. (new): The method of claim 69 wherein the producing step further comprises producing an arithmetic scalar condition state.

71. (new): The method of 70 wherein the storing step further comprises storing the arithmetic scalar condition state.

72. (new): The method of claim 71 wherein the storing step a latch is used for storing the instruction control signal for the instruction after the instruction has entered the execution state.

73. (new): The method of claim 72 wherein the producing step further comprises providing a Boolean combination of a present selected state with a previous state.